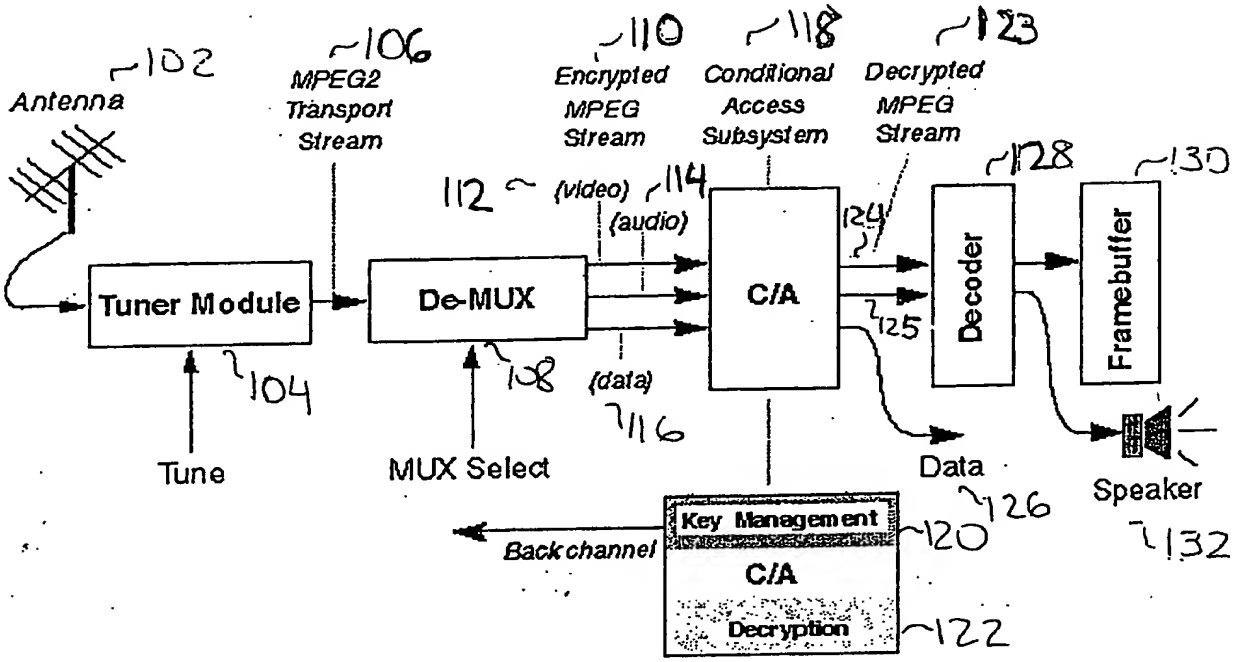
102
Antenna
106
MPEG2 Transport Stream
110
Encrypted MPEG Stream
118
Conditional Access Subsystem
123
Decrypted MPEG Stream
128
130
112
(video)
114
(audio)
124
125
Tuner Module
De-MUX
C/A
Decoder
Framebuffer
104
108
(data)
116
Tune
MUX Select
Data
126
Speaker
132
Backchannel
Key Management
120
C/A
Decryption
122

FIG. 1

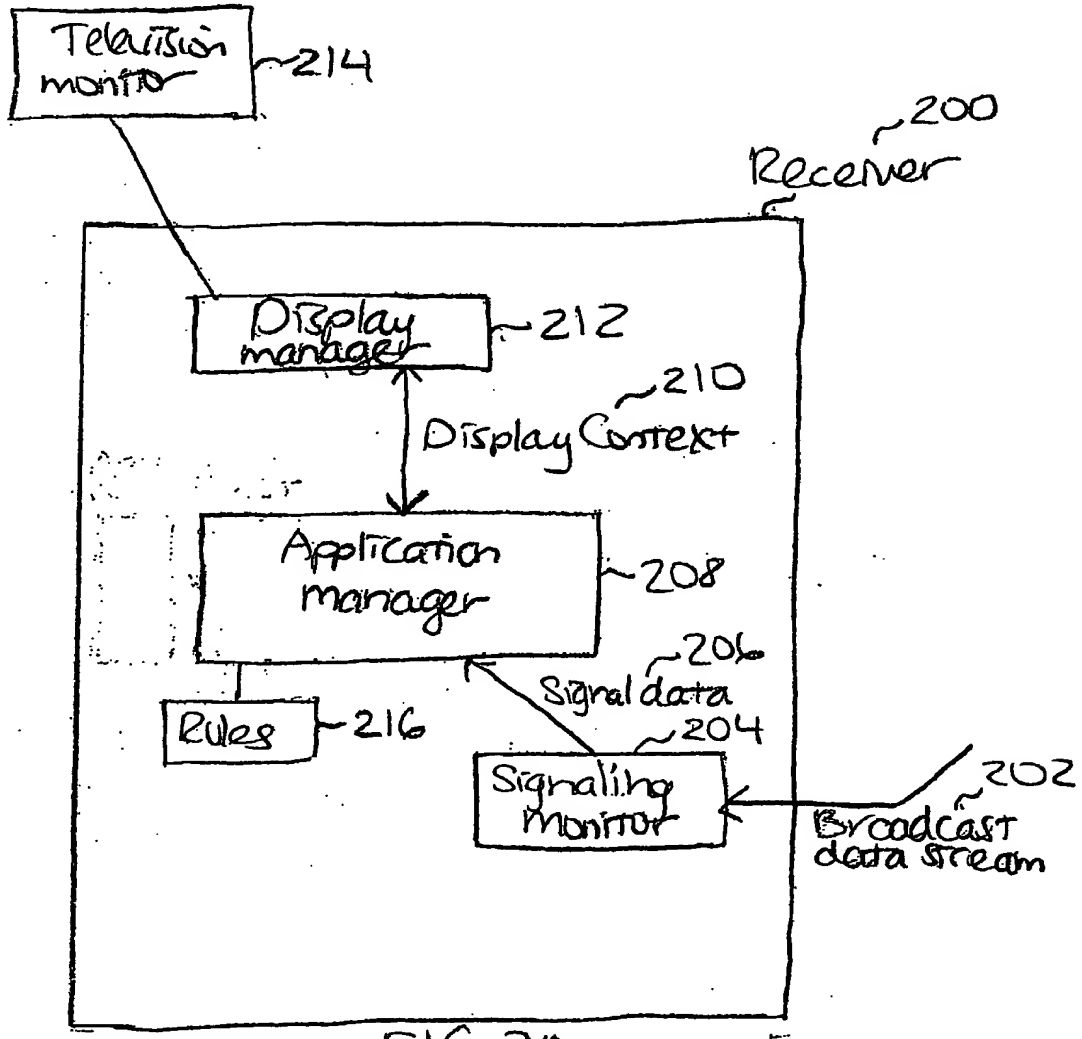

FIG. 2A

Rules 216

218 — 1. Application manager manages one or more applications

220 — 2. Only one application can be active at a time

222 — 3. Only one application can be displayed at a time

224 — 4. Only active applications are displayed

FIG. 2B

Signal Data ~302

App-present-in-data-stream: BOOLEAN	~304
Location (e.g., directory, filename)	~306
Authentication ~308	
Permissions ~310	

FIG. 3

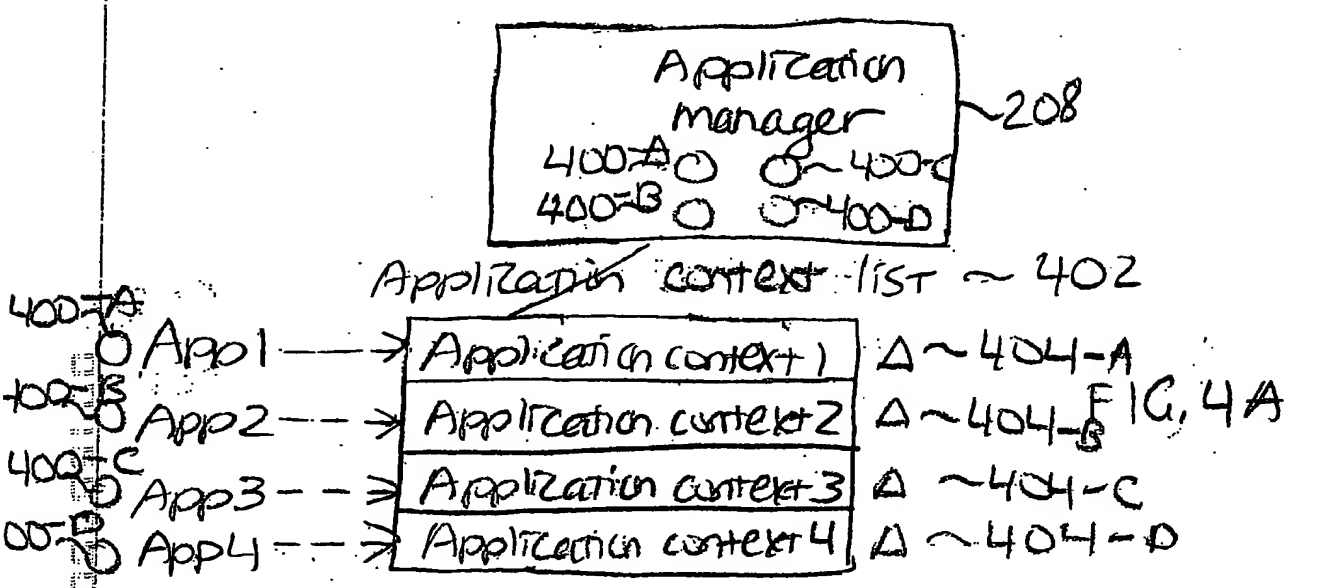

FIG. 4A

Application Context Δ ~404

Class loader identifier	406
Signal data identifier	408
Display context identifier	410
Application identifier	412
Application environment Object	414
Current application state	416

FIG. 4B

List of display contexts ~502

400-A App1 →	101	Context 1 ~506
400-B App2 →	102	Context 2
400-C App3 →	103	Context 3
400-D App4 →	104	Context 4

~504

FIG. 5A

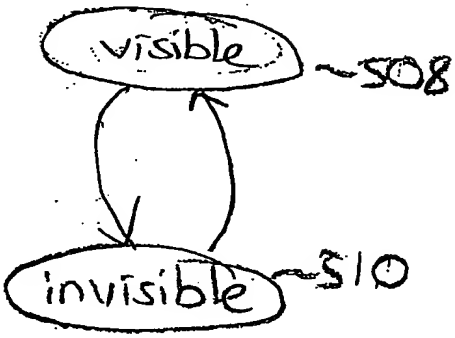

FIG. 5B

512

	0	1
516 ~ Application state	Active	Paused
514 ~ Display state	Visible	Invisible

FIG. 5C

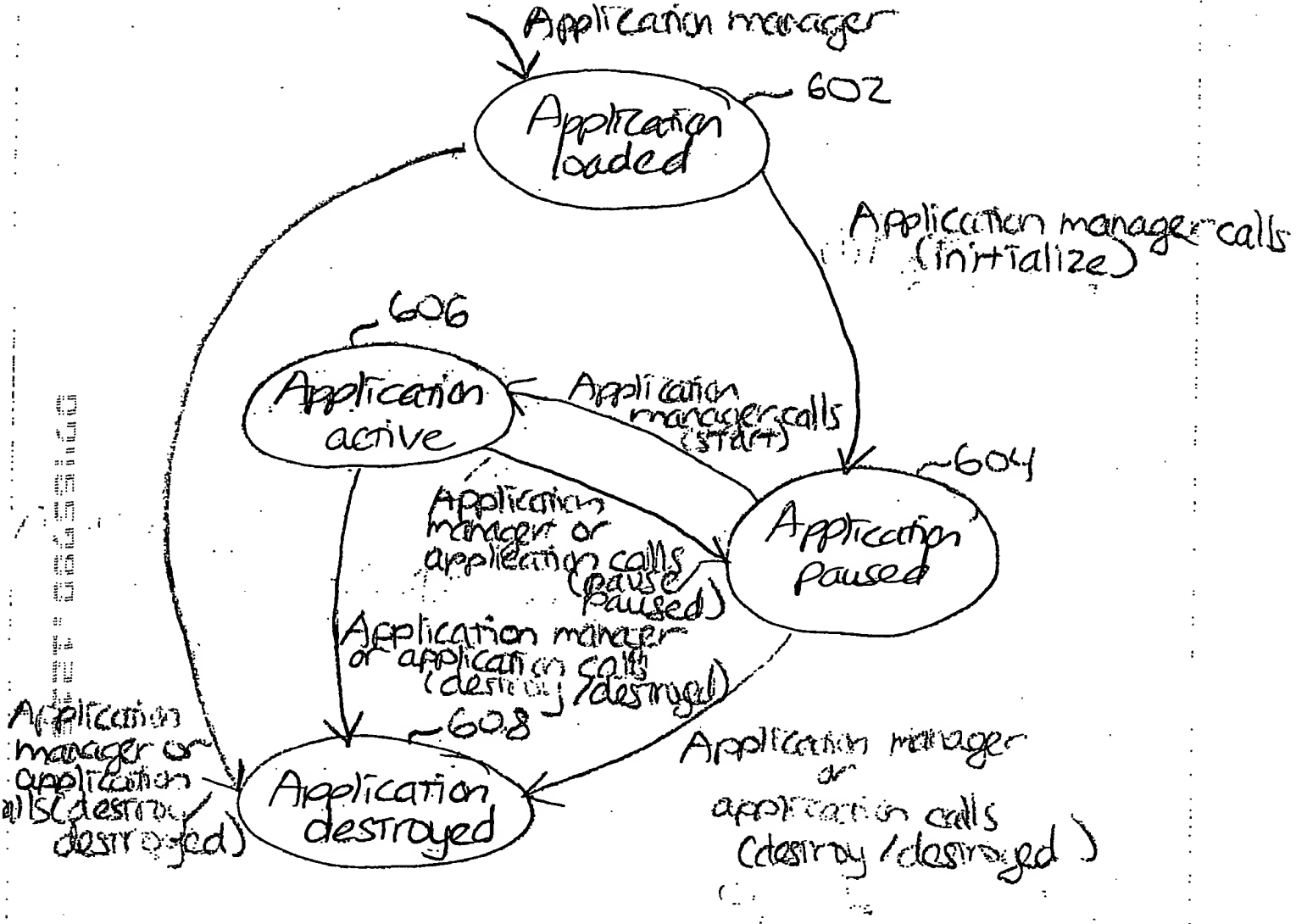
Application manager
602
Application loaded
Application manager calls (initialize)
606
Application active
Application manager calls (start)
604
Application paused
Application manager or application calls (pause / paused)
Application manager or application calls (destroy / destroyed)
608
Application destroyed
Application manager or application calls (destroy / destroyed)
Application manager or application calls (destroy / destroyed)

FIG. 6

Application Programming Interface (API)

~ 702

Interface application {

704 ~ public void init application (Application environment env); ~ 706
708 ~ public void start application();
710 ~ public void pause application();
712 ~ public void destroy application (boolean unconditional); ~ 714

}

to running state

FIG. 7

Application environment interface

```
                                    ~802
Interface Application-environment {
804 ~ public void destroyed ();
808 ~ public java.lang.Object getProperty (String key);   ~810
806 ~ public void paused ();
812 ~ public void resumeRequest ();
}
```

FIG. 8

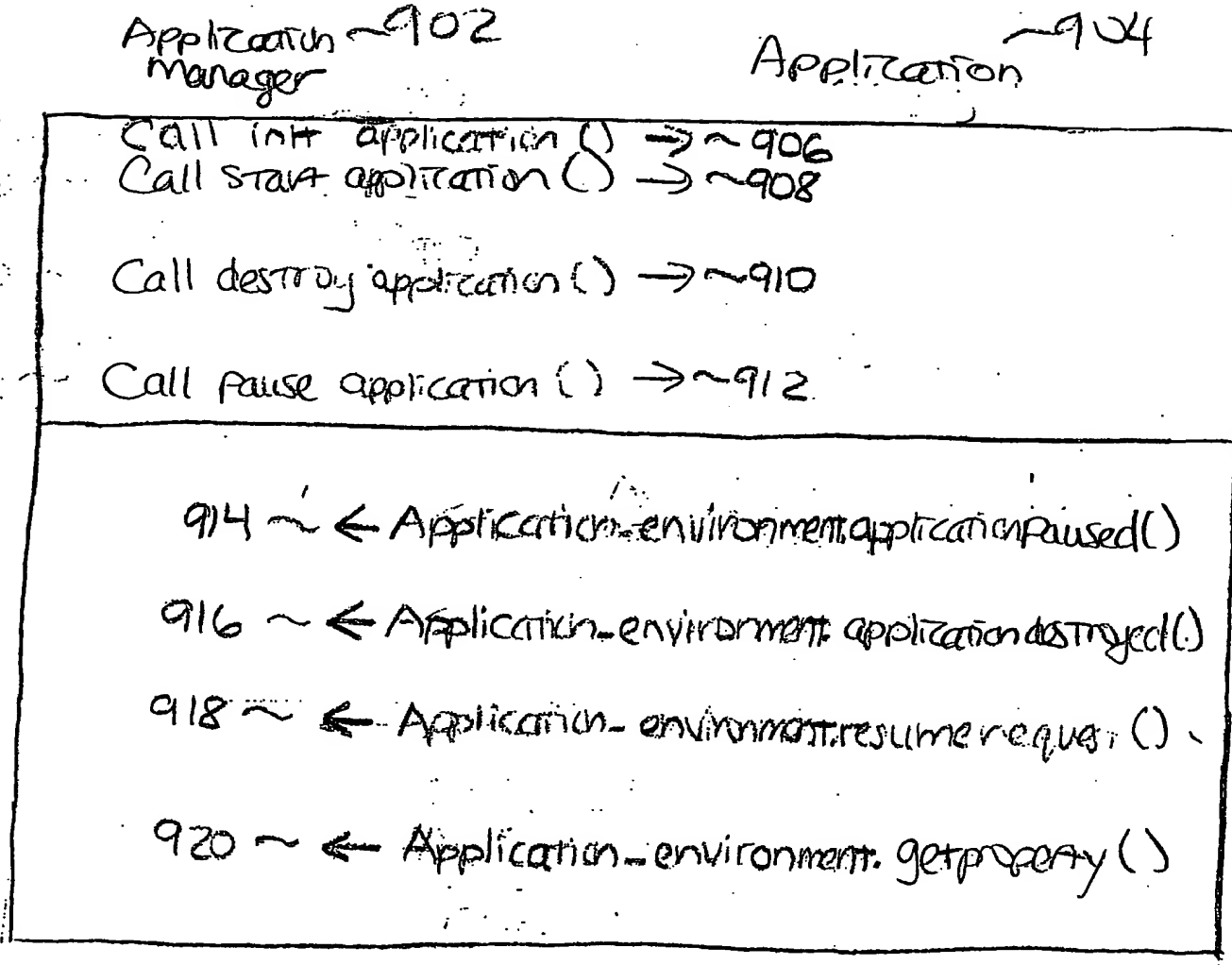
Application Manager ~902
Application ~904
Call init application () → ~906
Call start application () → ~908
Call destroy application () → ~910
Call pause application () → ~912
914 ~ ← Application_environment.applicationPaused()
916 ~ ← Application_environment.applicationDestroyed()
918 ~ ← Application_environment.resumeRequest()
920 ~ ← Application_environment.getProperty()

FIG. 9

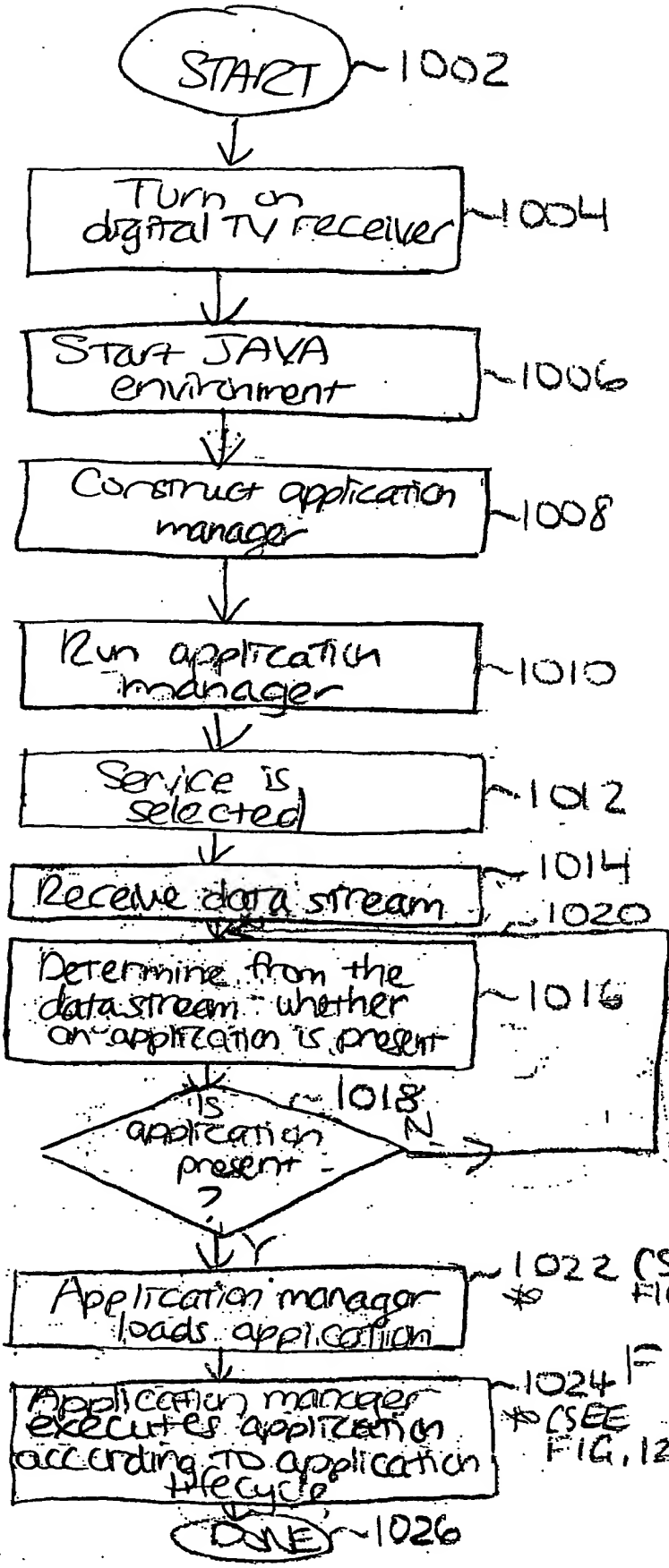
START
1002
Turn on digital TV receiver
1004
Start JAVA environment
1006
Construct application manager
1008
Run application manager
1010
Service is selected
1012
Receive data stream
1014
1020
Determine from the data stream whether an application is present
1016
Is application present ?
1018
N
Y
Application manager loads application
1022 (SEE FIG. 11)
Application manager executes application according to application lifecycle
1024 (SEE FIG. 12)
DONE
1026

FIG. 10

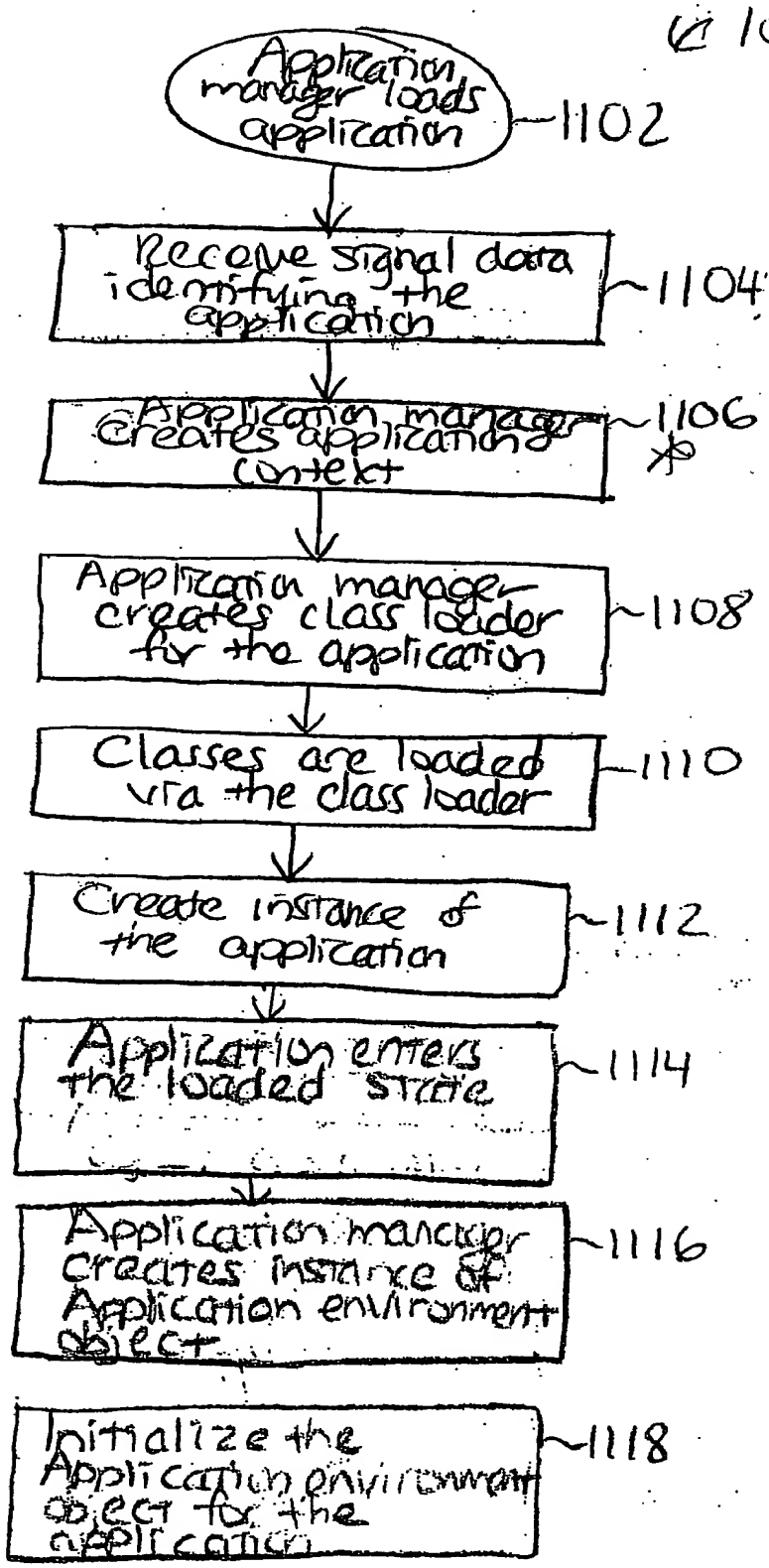
1022
Application manager loads application
1102
Receive signal data identifying the application
1104
Application manager creates application context
1106
Application manager creates class loader for the application
1108
Classes are loaded via the class loader
1110
Create instance of the application
1112
Application enters the loaded state
1114
Application manager creates instance of Application environment object
1116
Initialize the Application environment object for the application
1118

FIG. 11

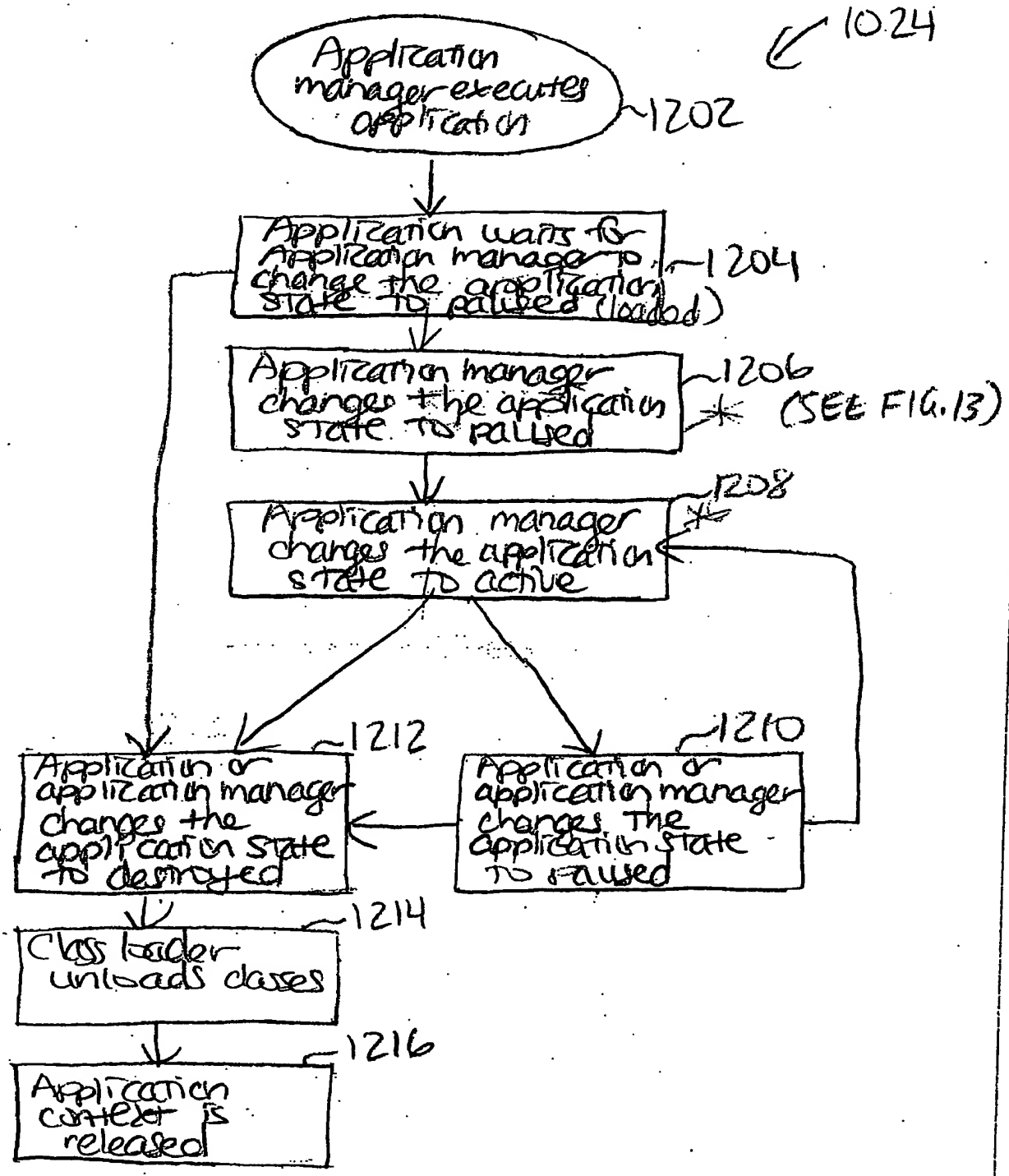
1024
Application manager executes application
1202
Application waits for Application manager to change the application state to paused (loaded)
1204
Application manager changes the application state to paused
1206
(SEE FIG. 13)
1208
Application manager changes the application state to active
1212
Application or application manager changes the application state to destroyed
1210
Application or application manager changes the application state to paused
1214
Class loader unloads classes
1216
Application context is released

FIG. 12

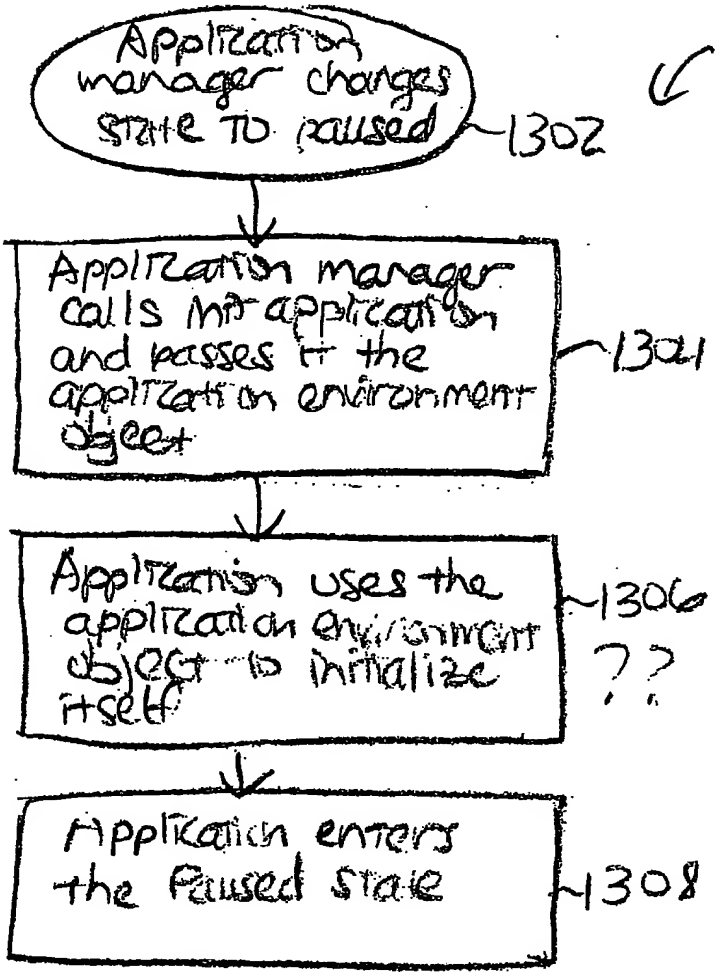
Application manager changes state TO paused
1302
1206
Application manager calls init application and passes it the application environment object
1304
Application uses the application environment object to initialize itself
1306
??
Application enters the Paused state
1308

FIG. 13

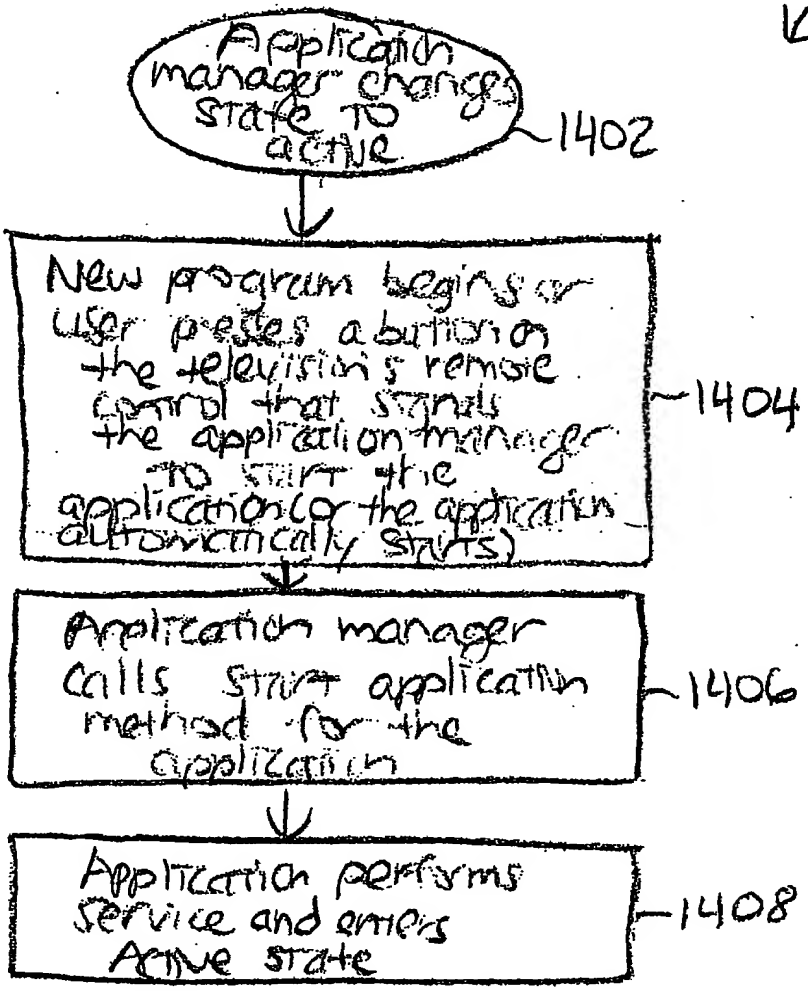
1208
Application manager changes state to active
1402
New program begins or user presses a button on the television's remote control that signals the application manager to start the application (or the application automatically starts)
1404
Application manager calls start application method for the application
1406
Application performs service and enters Active state
1408

FIG. 14

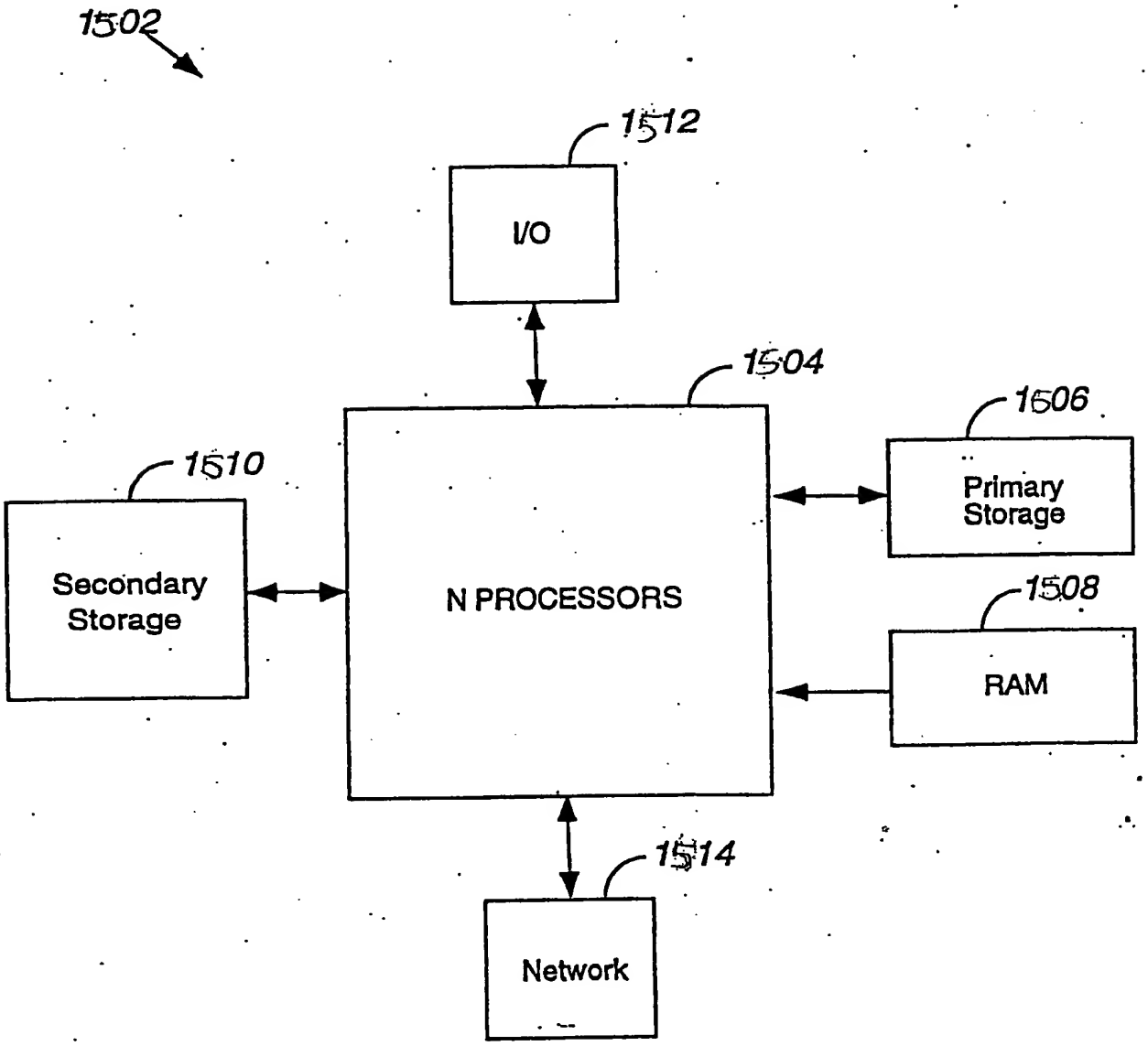
1502
1512
I/O
1504
N PROCESSORS
1506
Primary Storage
1510
Secondary Storage
1508
RAM
1514
Network

Figure 15

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☑ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.